



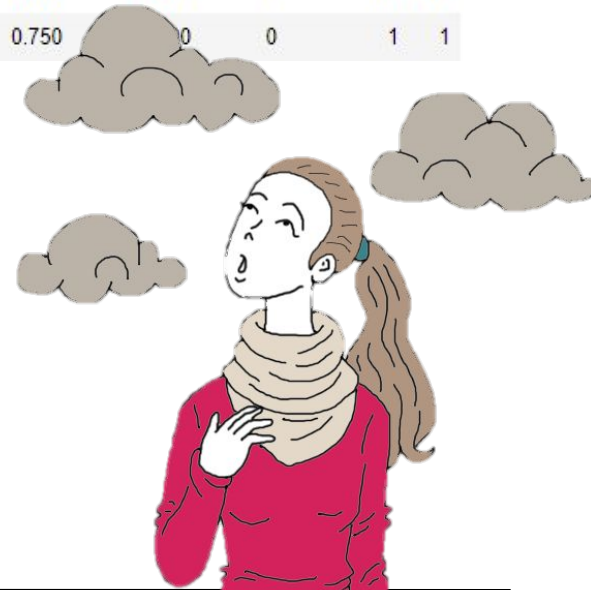
Predicció de lloguer de bicicletes

Adrià Carrasquilla
Jan Moros
Grup 202

Entrega anterior

Primer contacte amb la BD

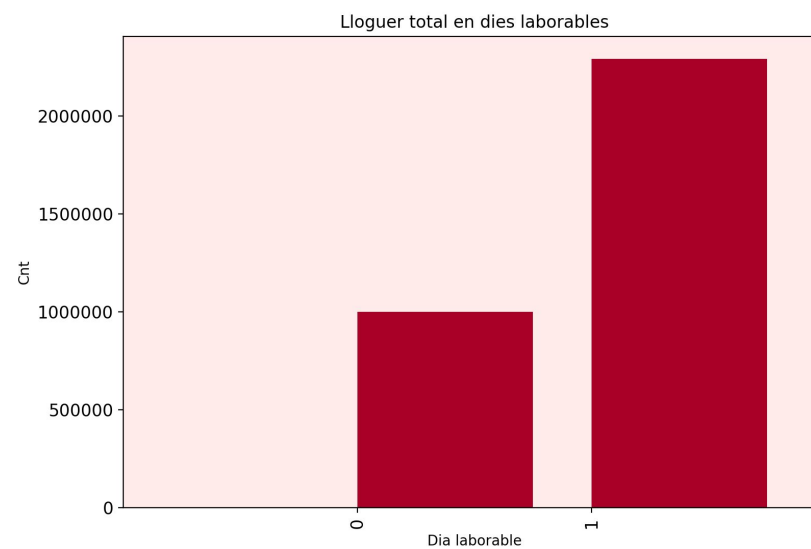
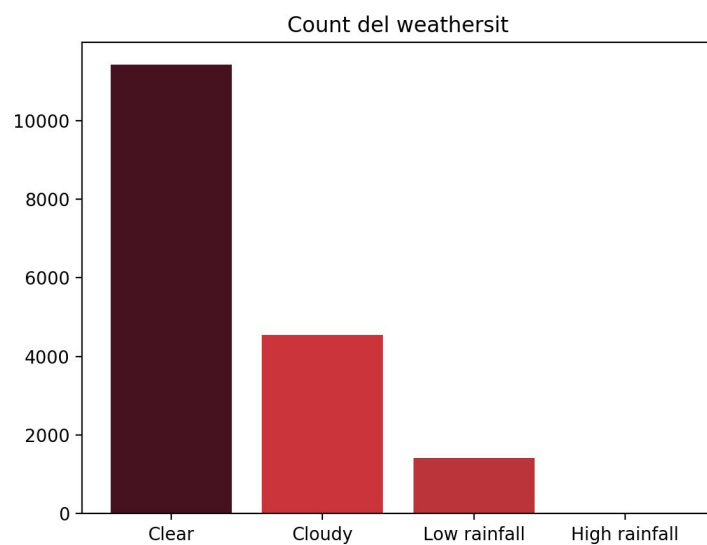
	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.240	0.288	0.810	0.000	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.220	0.273	0.800	0.000	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.220	0.273	0.800	0.000	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.240	0.288	0.750	0.000	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.240	0.288	0.750	0.000	0	1	1



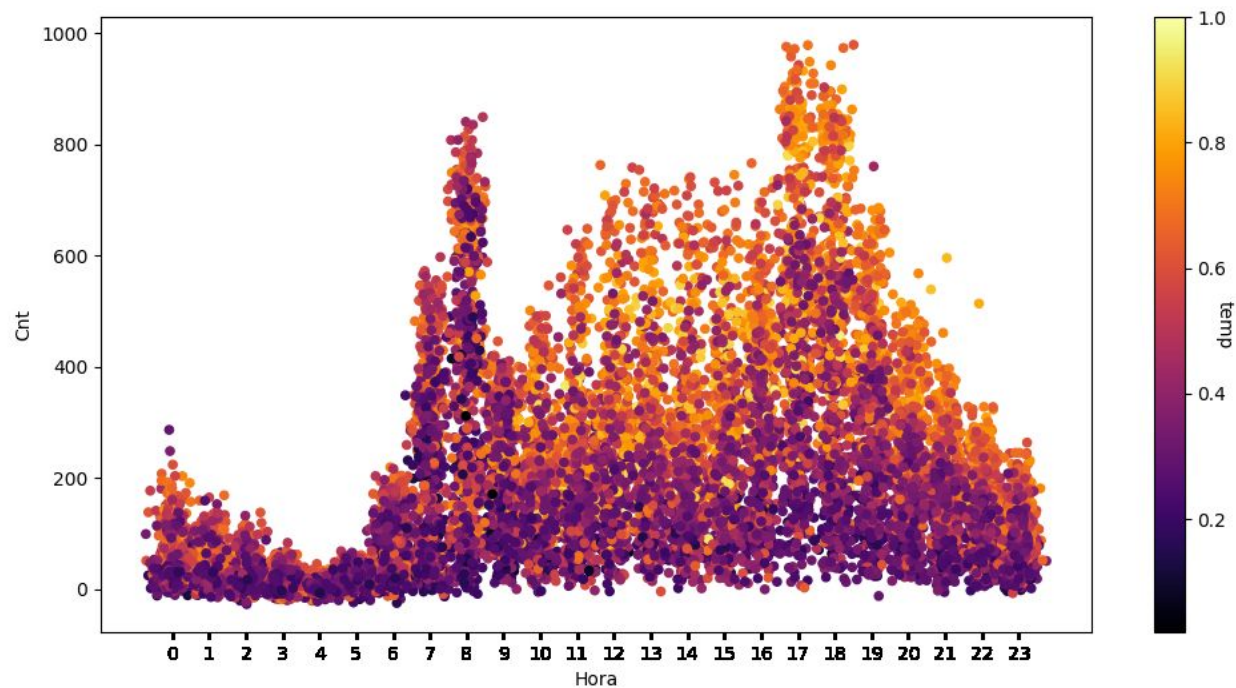
Segona Entrega

- Anàlisi exhaustiu Dataset
- Culminació regresor lineal
- Implementació descens del gradient

Anàlisi Dataset



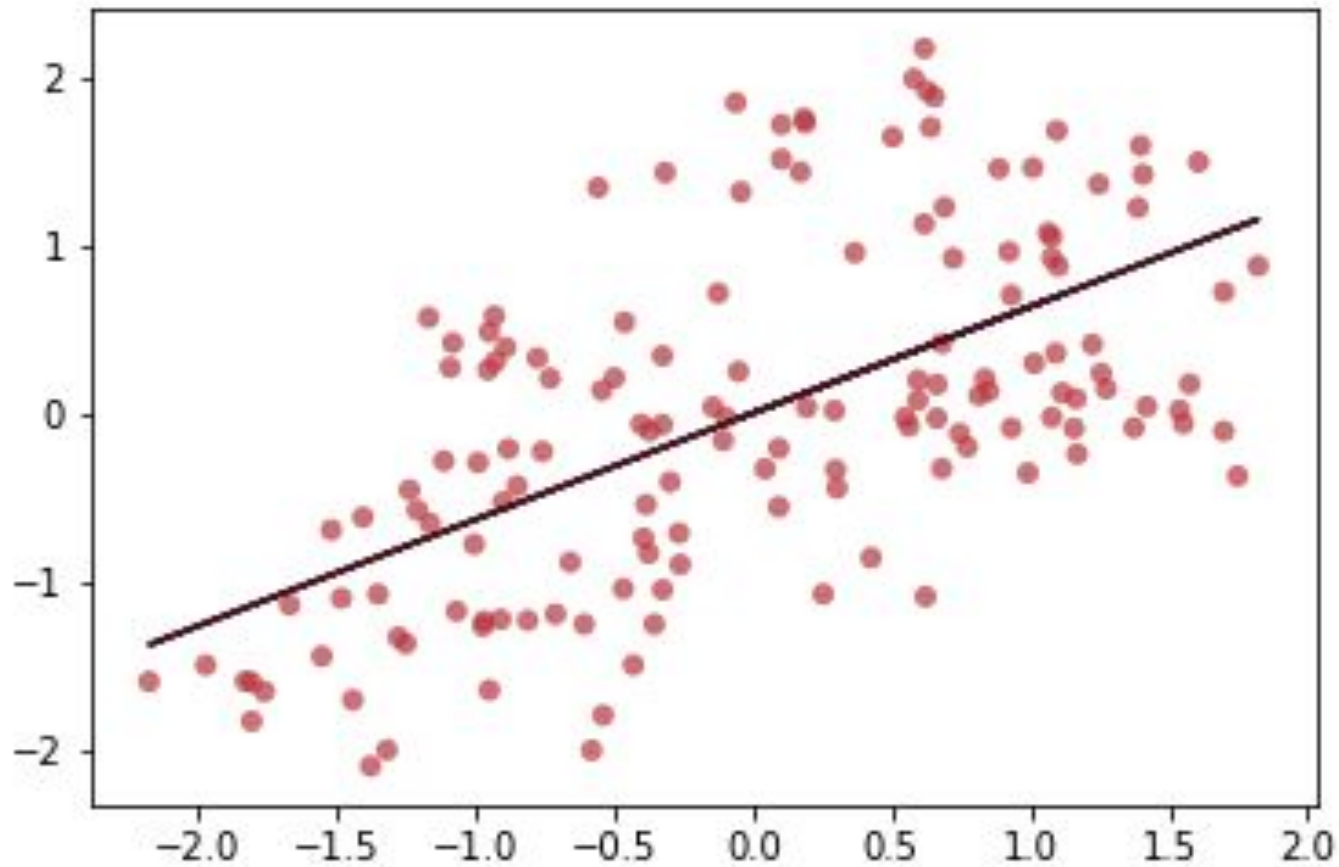
Anàlisis Dataset



Regressor Lineal

Atribut	Mean Square Error	R2 score
season	0.8873	0.1613
year	0.76	0.2815
month	1.0292	0.027
holiday	1.0781	-0.019
weekday	1.0544	0.0034
workingday	1.07	-0.0115
weathersit	0.9681	0.085
temp	0.5657	0.4653
hum	1.063	-0.0047
windspeed	1.0216	0.0344

Regressor Lineal

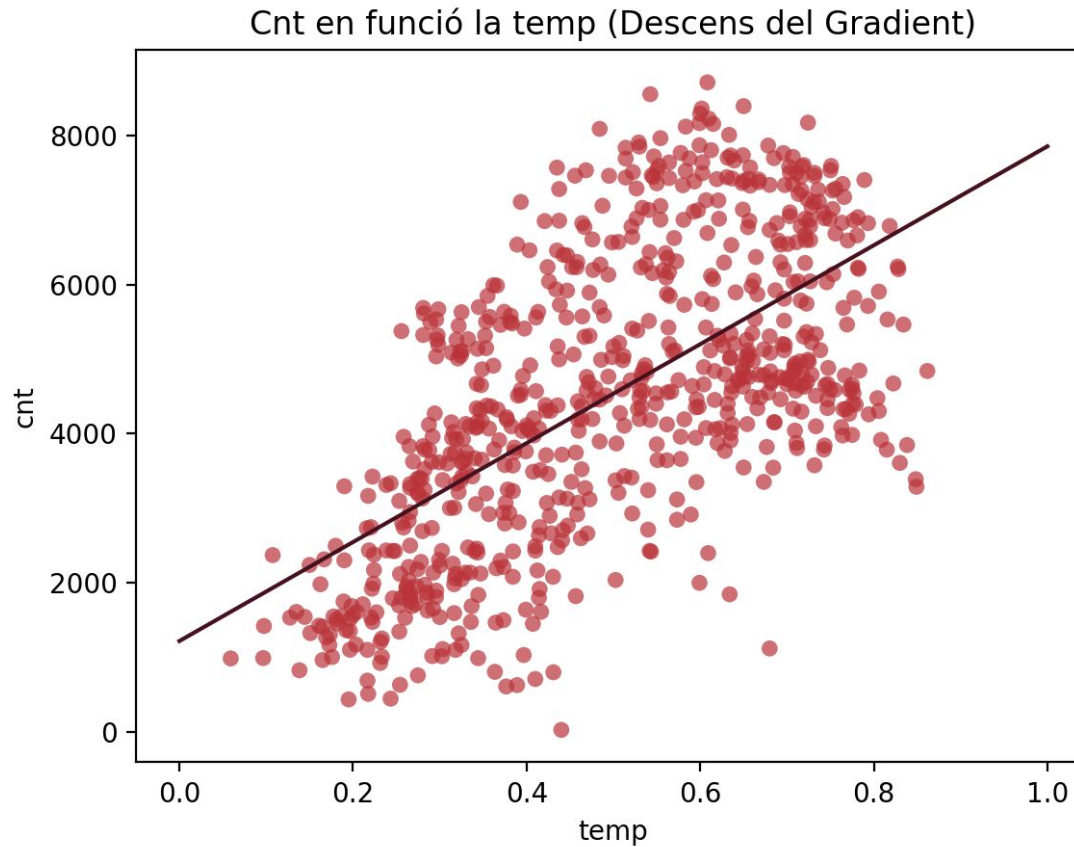


Descens del gradient

Implementació de la funció:

```
def Regressor(x, y, arrayTheta, max_iter, epsilon, aplha):  
    i = 1  
    millora=epsilon+1  
    costAnt=0  
    while( i < max_iter and millora > epsilon):  
        predict = np.dot(x, arrayTheta)  
        loss = predict - y  
        cost = np.sum(loss ** 2) / (2 * x.shape[0])  
        millora = abs(costAnt-cost)  
        costAnt = cost  
        gradient = np.dot(x.T, loss) / x.shape[0]  
        print("Iter: "+str(i)+" Cost: "+str(cost))  
        arrayTheta = arrayTheta - alpha * gradient  
        i+=1  
    return arrayTheta
```


Descens del gradient



Descens del gradient

Anàlisi paràmetres:

- Alpha
- Max_iterations
- Epsilon

Estudi amb diferents funcions polinomials

